Fact Set

Problem Description

Use the movies metadata file to predict the genres field.

Data Set

Downloaded from here: https://www.kaggle.com/rounakbanik/the-movies-dataset

Columns to be used from the Dataset

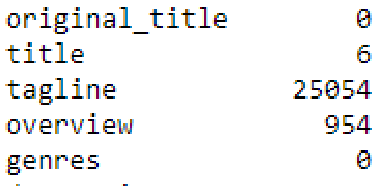
- Title
- Original Title
- Tagline
- Overview

Shape of the Dataset

In Meta_data file we have 45466 rows and 24 columns(From these columns we need to use only the columns given above)

After taking those columns in to account the shape of the data set is 45466 rows and 5 columns

Null Values in the Dataset



```
original_title        0
title                 6
tagline           25054
overview            954
genres                0
```

From the above we see that the tagline column is having 50% of the Null Values and overview is having 2% of Null Values.

Step Taken

- Tried Joining two columns(tagline and overview) and tried to fill the Null Values but some of the rows are still having the Null Values So Decided to drop the Null Values after joining.
- Similarly original title and title column are compared and seen that some of the other languages titles were also there. Ex : - æ‘‡å•Šæ‘‡ï¼Œæ‘‡åˆ°å¤–å©†æ¡¥ ------→original title name
  Shanghai Triad-------------------------------- →title
- Due to this using Text-Blob Library in python will detect and convert the text in another language
- But due to Http Error: Too Many Request I was not using the two columns for my analysis.
- I used only overview column to predict the genres

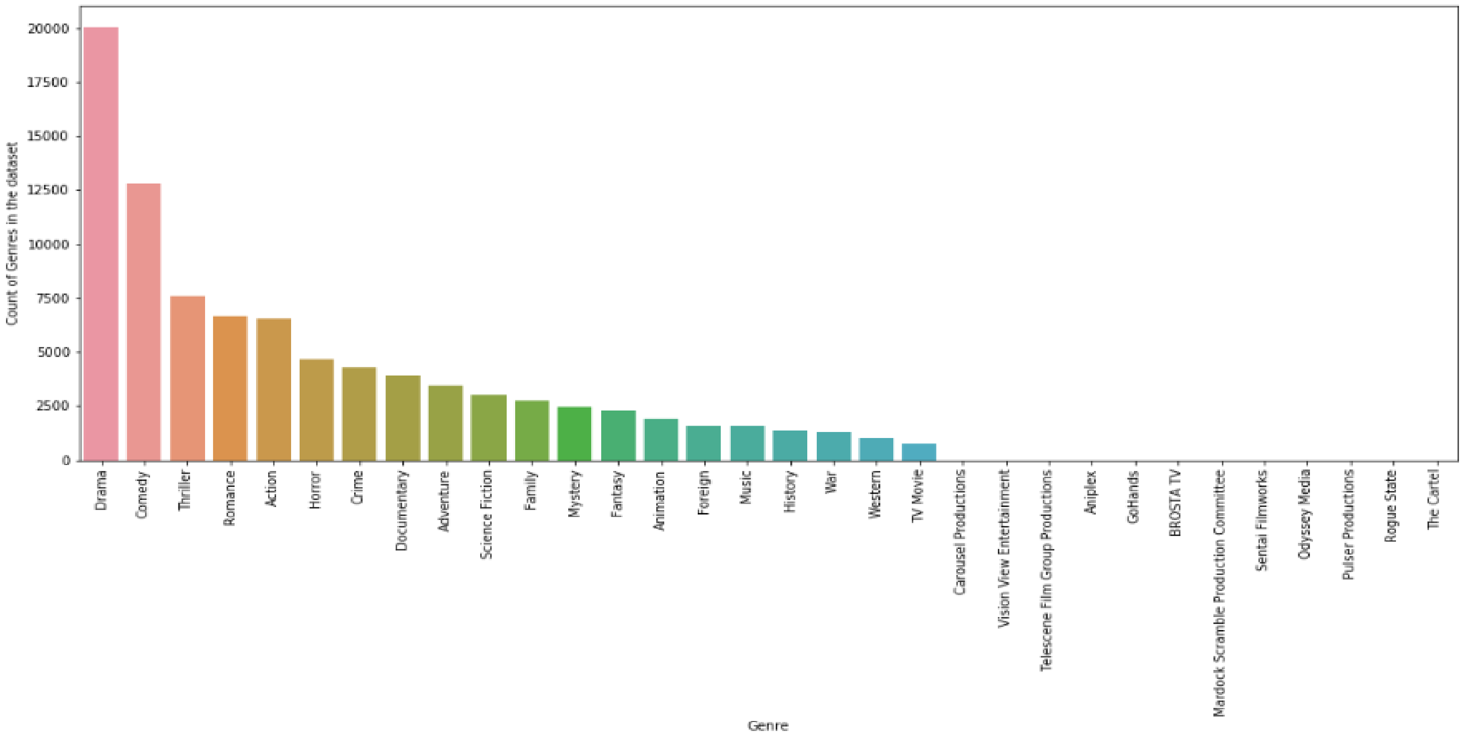
Target Column

Genres column which we want to predict. But the Labels is in a dictionary format and we have to extract only the genres from that dictionary.

Genres labels: [{'id': 35, 'name': 'Comedy'}, {'id': 18, 'name': 'Drama'}, {'id': 10749, 'name': 'Romance'}]

After Extracting: [‘Comedy’,’Drama’,’Romance’]

Count of Genres in the dataset

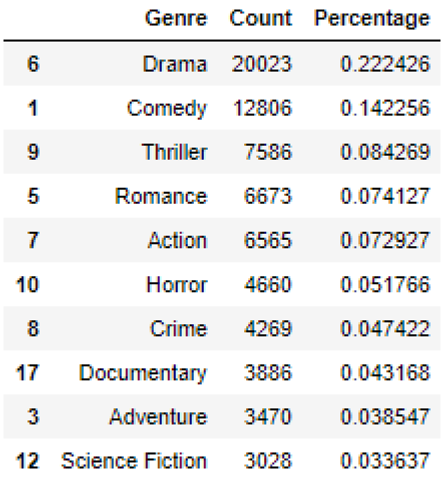


From above we see that Drama, Comedy, Thriller, Romance etc., is the genres of most of the movies in the dataset.

But we can see some of the genres Carousel Production, Vision View Entertainment, Aniplex, Go hands etc., which has only one record so decided to remove it. Let see Percentage wise Genres distribution.

Percentage wise Distribution of Genres

Top 10 Genres



	Genre	Count	Percentage
6	Drama	20023	0.222426
1	Comedy	12806	0.142256
9	Thriller	7586	0.084269
5	Romance	6673	0.074127
7	Action	6565	0.072927
10	Horror	4660	0.051766
8	Crime	4269	0.047422
17	Documentary	3886	0.043168
3	Adventure	3470	0.038547
12	Science Fiction	3028	0.033637

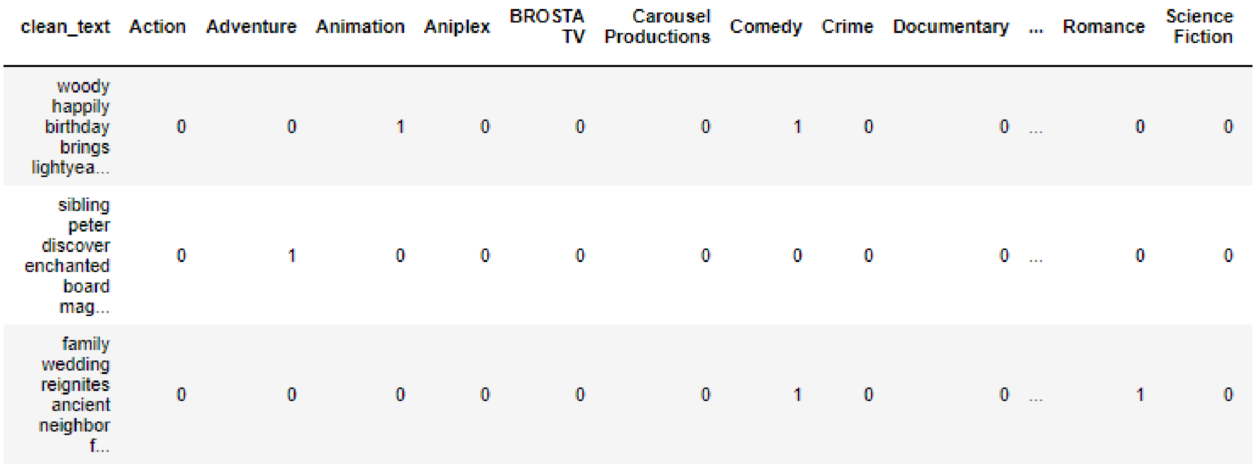
We can see the Drama Genres is 22%, Comedy 14%, Thriller 8% and so on are the top genres for most of the movies in the dataset.

But we know that one movie will contain multiple label as [Action, Adventure, Drama] in the Genre Column we need to split these things as separate columns so that Machine Learning Algorithms Learn from the text to predict the genres.

For Doing this we use of Multilabel Binarizer .

Multilabel Binarizer

From the sklearn library in python we use this library called Multilabel Binarizer where it will convert list of genres as different column as shown below.



clean_text	Action	Adventure	Animation	Aniplex	BROSTA TV	Carousel Productions	Comedy	Crime	Documentary	...	Romance	Science Fiction
woody happily birthday brings lightyea...	0	0	1	0	0	0	1	0	0	...	0	0
sibling peter discover enchanted board mag...	0	1	0	0	0	0	0	0	0	...	0	0
family wedding reignites ancient neighbor f...	0	0	0	0	0	0	1	0	0	...	1	0

In this we are using the column clean text and the text inside that column to predict the Remaining column which is the Action, Adventure, Animation etc.

You can see some value in the Animation, Comedy etc. which represent that for a particular movie overview it belongs to Action and not a Aniplex For first text.

For predicting this we use the text column from the overview but overview is the raw data where machine Learning models are not learned by text but with numbers. So, we need to convert these texts to a Numerical Data so that we can pass it to the Model For learning. But How? Let's see…

Cleaning the text

Some of the basic idea of removing unwanted symbols, letters, and word which is not useful in the prediction of Genre.

- Removing Tags - Our text often contains unnecessary content like HTML tags, which do not add much value when analyzing text.
- Removing Accented Characters - A simple example would be converting **é** to **e**.
- Expanding Contractions - Examples would be, ***do not*** to ***don't*** and ***I would*** to ***I'd***.
- Removing Special Characters - #@$%^*()?: removing these Symbols
- Stemming and Lemmatization - ***WATCH**ES*, ***WATCH**ING*, and ***WATCH**ED*. They have the word root stem ***WATCH*** as the base form.
- Remove Stop Words - Words like ***a***, ***an***, ***the***, ***and*** so on are considered to be stopwords.

For Example :-

Raw Data → Led by Woody, Andy's toys live happily in his ...

Cleaned Data → led woody andy toy live happily room andy birt...

But here we are Still Cleaning the data once the cleaning is done make use of that text and make a column using a word in the text. For Doing this we use of Count-Vectorizer and Tf-Idf Vectorizer.

Lets Discuss About Brief and Example with our Analysis.

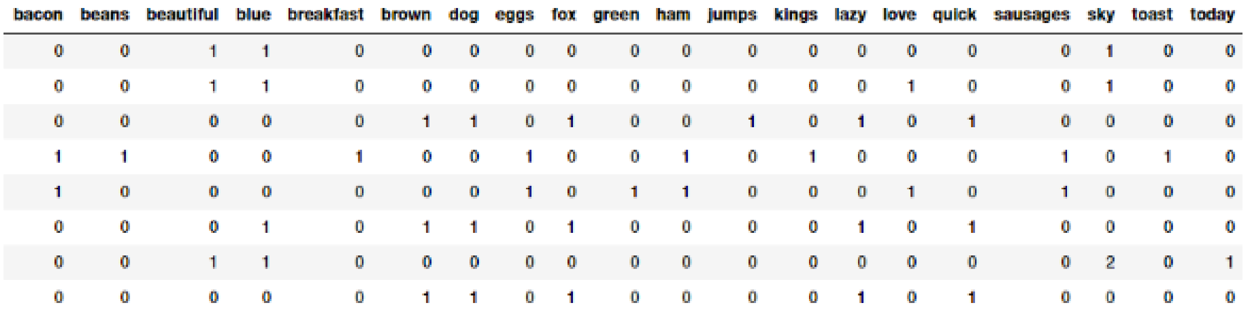
Bag of Words Features or Columns using Count Vectorizer

Consider a Corpus C of D documents {d1,d2…..dD} and N unique tokens extracted out of the corpus C. The N tokens (words) will form a dictionary and the size of the bag-of-words matrix M will be given by D X N. Each row in the matrix M contains the frequency of tokens in document D(i).

- Let us understand this using a simple example.
- D1: He is a lazy boy. She is also lazy.
- D2: Smith is a lazy person.

The dictionary created would be a list of unique tokens in the corpus = [‘He’,’She’,’lazy’,’boy’,’Smith’,’person’]

Here, D=2, N=6 For example,

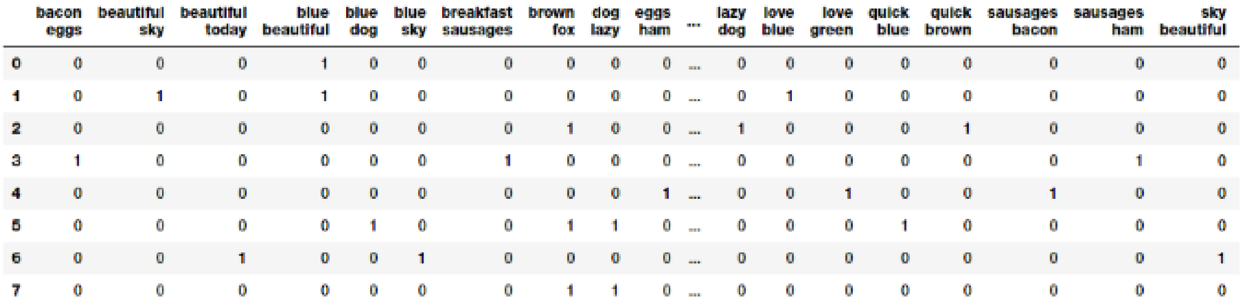


bacon	beans	beautiful	blue	breakfast	brown	dog	eggs	fox	green	ham	jumps	kings	lazy	love	quick	sausages	sky	toast	today
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	1	0	0	0	0
1	1	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	1	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	0	0
0	0	0	1	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1
0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	0

For every word it create a column and gives either 1 or 0 if that particular word exist in that document.

Bag of N-Grams Model

Nothing Special About this the Column Created above will combination of two word and gives one if two word in combinable arrived else 0. For example,



	bacon eggs	beautiful sky	beautiful today	blue beautiful	blue dog	blue sky	breakfast sausages	brown fox	dog lazy	eggs ham	...	lazy dog	love blue	love green	quick blue	quick brown	sausages bacon	sausages ham	sky beautiful
0	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	...	1	0	0	0	1	0	0	0
3	1	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	1	...	0	0	1	0	0	1	0	0
5	0	0	0	0	1	0	0	1	1	0	...	0	0	0	1	0	0	0	0
6	0	0	1	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1	1	0	...	0	0	0	0	0	0	0	0

By giving these numerical values to the Machine Learning Modelit will learn from the values to predict the genres.

Tf-Idf Features (Term Frequency – Inverse Document Frequency)

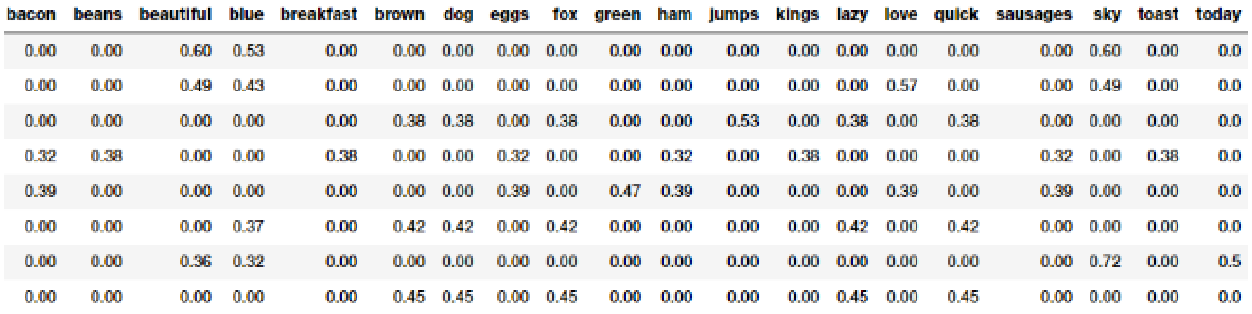
This is another method which is based on the frequency method but it is different to the bag-of-words approach in the sense that it takes into account not just the occurrence of a word in a single document (or tweet) but in the entire corpus.

Tf-Idf works by penalizing the common words by assigning them lower weights while giving importance to words which are rare in the entire corpus but appear in good numbers in few documents.

Let’s have a look at the important terms related to Tf-Idf:

- TF = (Number of times term t appears in a document)/(Number of terms in the document)
- IDF = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.
- TF-IDF = TF*IDF

Example

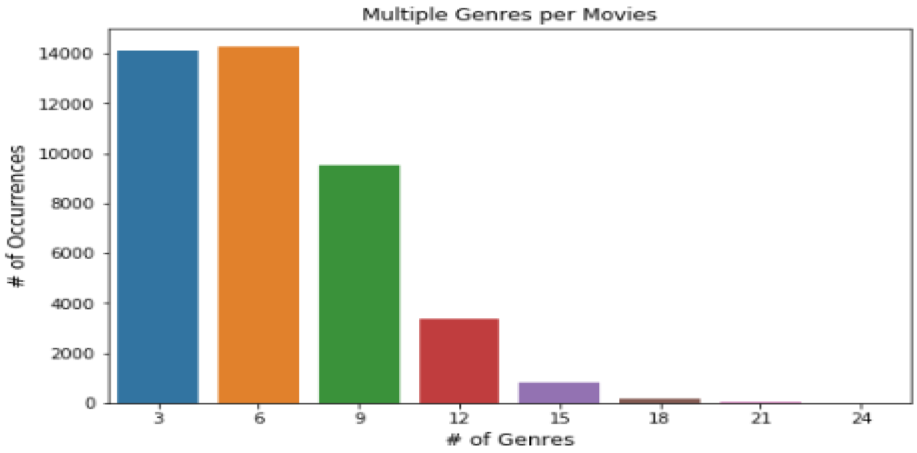


bacon	beans	beautiful	blue	breakfast	brown	dog	eggs	fox	green	ham	jumps	kings	lazy	love	quick	sausages	sky	toast	today
0.00	0.00	0.60	0.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.00	0.0
0.00	0.00	0.49	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.57	0.00	0.00	0.49	0.00	0.0
0.00	0.00	0.00	0.00	0.00	0.38	0.38	0.00	0.38	0.00	0.00	0.53	0.00	0.38	0.00	0.38	0.00	0.00	0.00	0.0
0.32	0.38	0.00	0.00	0.38	0.00	0.00	0.32	0.00	0.00	0.32	0.00	0.38	0.00	0.00	0.00	0.32	0.00	0.38	0.0
0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.39	0.00	0.47	0.39	0.00	0.00	0.00	0.39	0.00	0.39	0.00	0.00	0.0
0.00	0.00	0.00	0.37	0.00	0.42	0.42	0.00	0.42	0.00	0.00	0.00	0.00	0.42	0.00	0.42	0.00	0.00	0.00	0.0
0.00	0.00	0.36	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.72	0.00	0.5
0.00	0.00	0.00	0.00	0.00	0.45	0.45	0.00	0.45	0.00	0.00	0.00	0.00	0.45	0.00	0.45	0.00	0.00	0.00	0.0

You can see that the values in each cell is in float ,frequency across the documents and also the in his own document also.

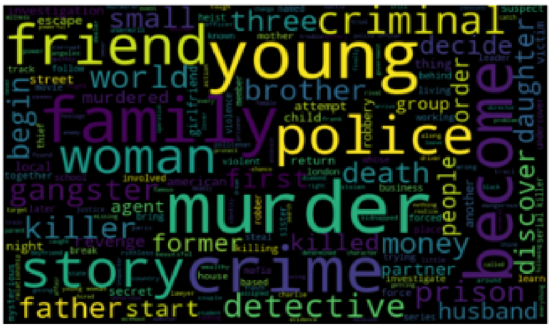
After creating these features lets see the most words which are representing all genres…But we know that the Genres we have around 32 and some of the genres are having one record so we removed them. But we some of Genres having a words that are helpful in predicting those genres.

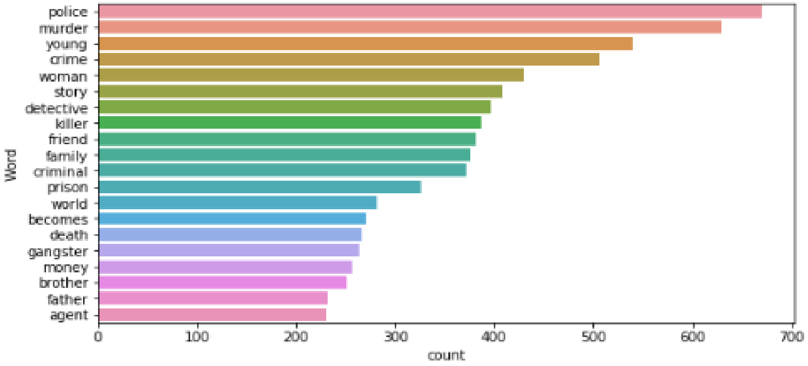
Multiple Genres in every Movies



We can most of the movies are having Multiple Genres. Most of the movies has 3 genres some have 6 genres and 9 genres so on.

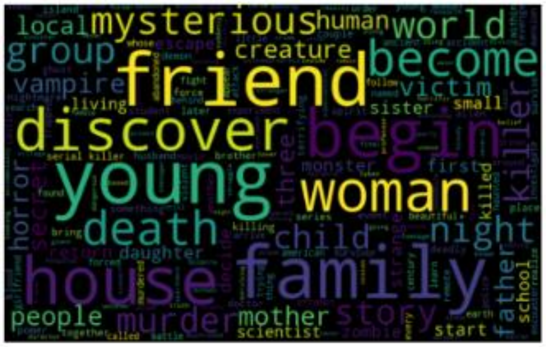
Crime

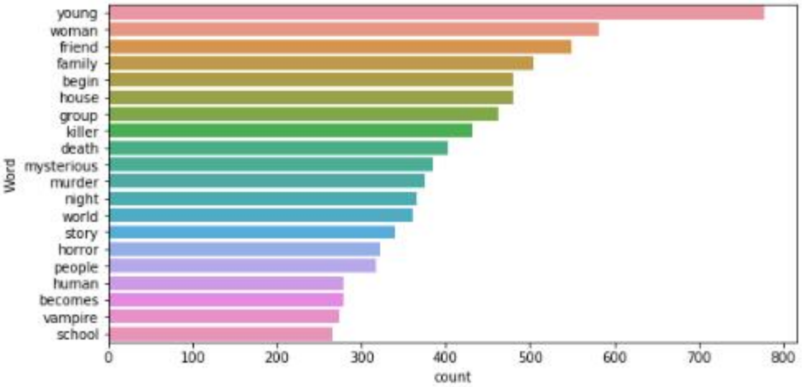




We see that for describing the crime movies the words which are in bold and the frequency of the word shown in the right. In above we can see word criminal, murder, crime, police, escape and detective are the words from the data is explaining about the Crime Genres.

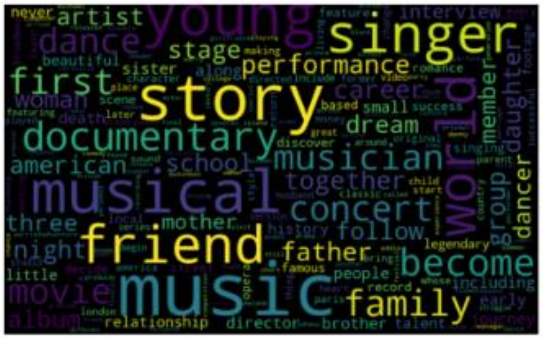
Horror

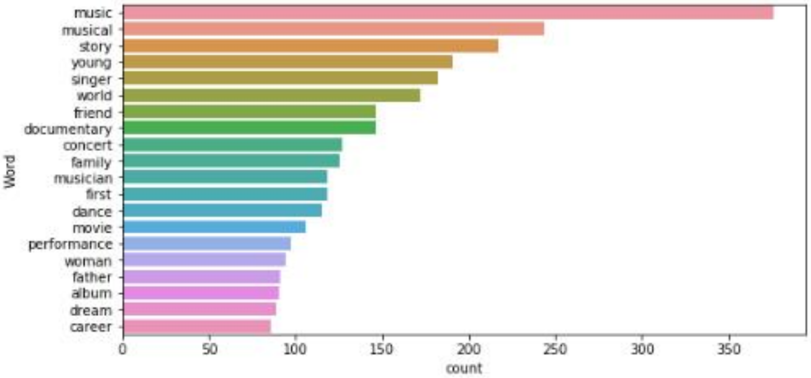




In the above we see that for Horror Movies there will be a Death, Creature, Mysterious, Vampire, Killing, Zombie are some of the word representing the Horror Movies From the given data

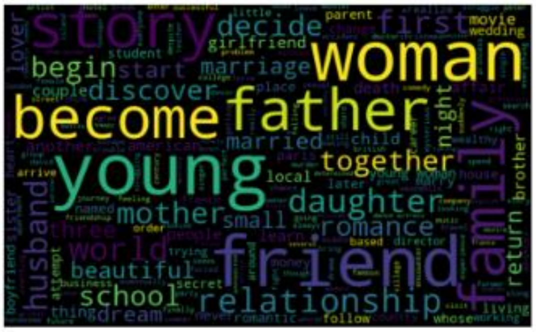
Music

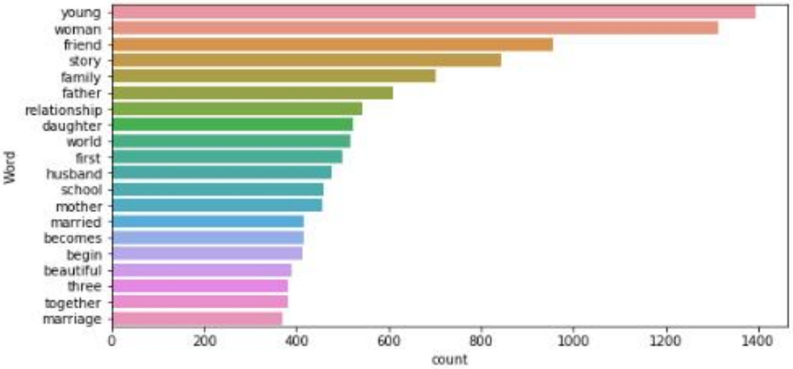




We can see that Music, Musical, Singer, concert , Musician, Dance, Album are some of the words which are representing the Music genres.

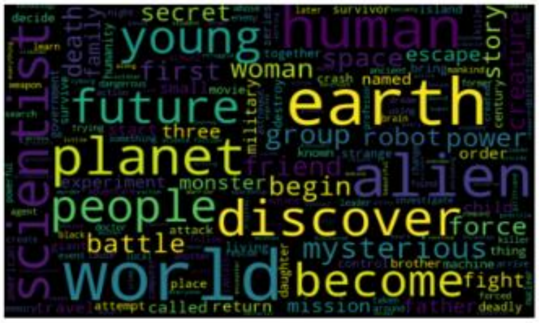
Romance

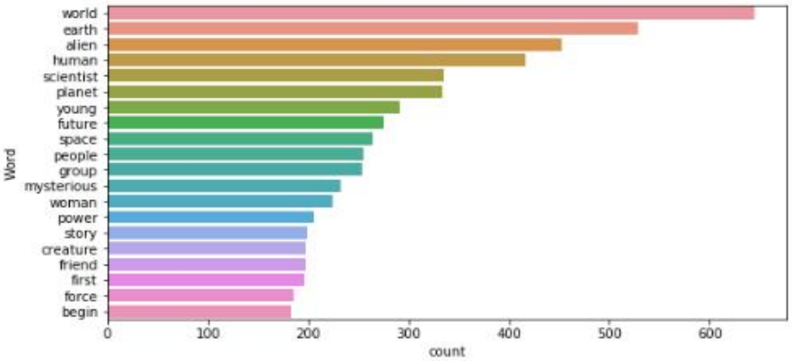




In this we can see Young,Family,Relationship,Love,Marriage,Beautiful are some of the keywords that are helpful in predicting the Romance .

Science Fiction





We see that word World,Earth,Alien,Human,Scientist,Planet,Future,Space are some of the words that are going to help in predicting the Science Fiction Movies from the given data.

For Detail Description of Words see the Code

Modeling

Machine Learning Models Tried

- Multi-Nomial Naïve Bayes – Gives a Better Result in predicting the Genres
- Linear SVC - Gives a Better Result in predicting the Genres
- Logistic Regression - Gives a Better Result in predicting the Genres
- Decision Tree – Not a Good Results

Future Analysis

- Will fetch an accurate word or good representation word for each genre in a list and tried to use only that words while training the model and predicting the Genres.
- We can try use of a recommendation engine for predicting these genres.
- Will try to use a balanced set of genres along with the text so that the model should not biased with one genres.